



US005802373A

United States Patent [19]

Yates et al.

[11] Patent Number: **5,802,373**[45] Date of Patent: **Sep. 1, 1998**[54] **METHOD FOR PROVIDING A PIPELINE INTERPRETER FOR A VARIABLE LENGTH INSTRUCTION SET**[75] Inventors: **John S. Yates, Needham; Stephen C. Root, Westboro, both of Mass.**[73] Assignee: **Digital Equipment Corporation, Maynard, Mass.**[21] Appl. No.: **592,982**[22] Filed: **Jan. 29, 1996**[51] Int. Cl.⁶ **G06F 9/45**[52] U.S. Cl. **395/705**[58] Field of Search **395/705, 707, 395/706**[56] **References Cited****U.S. PATENT DOCUMENTS**

5,287,490	2/1994	Sites .	
5,301,325	4/1994	Benson .	
5,307,492	4/1994	Benson .	
5,307,504	4/1994	Robinson et al. .	
5,317,740	5/1994	Sites .	
5,339,238	8/1994	Benson .	
5,339,422	8/1994	Brender et al. .	
5,428,786	6/1995	Sites .	
5,432,795	7/1995	Robinson .	
5,450,575	9/1995	Sites .	
5,507,030	4/1996	Sites .	
5,574,887	11/1996	Fitch	395/500
5,574,927	11/1996	Scandin	395/800
5,598,560	1/1997	Benson .	

OTHER PUBLICATIONS

"Binary Translation", Richard L. Sites et al., Digital Technical Journal, pp. 137-152.

"Adaptive Systems for the Dynamic Run-Time Optimization of Programs", G. Hansen, Carnegie-Mellon University, Mar., 1974, pp. 1-171.

"Data Structures and Algorithms", A. Aho et al., Computer science and Information Processing, pp. 110-117.

"Compiler Design Theory", P. Lewis, II et al., pp. 559-568.
 "Method and Apparatus for Direct Conversion of Programs in Object Code Form Between Different Hardware Architecture Computer Systems", J. Goettelmann et al., pp. 3-37.
 "Data Flow Analysis for 'Intractable' Imbedded System Software", H. Johnson, pp. 109-115.

(List continued on next page.)

Primary Examiner—Emanuel Todd Voeltz*Assistant Examiner*—Peter J. Corcoran, III*Attorney, Agent, or Firm*—Diane C. Drozenski; Ronald C. Hudgens[57] **ABSTRACT**

A computer system for executing a binary image conversion system which converts instructions from a instruction set of a first, non native computer system to a second, different, native computer system, includes a run-time system which in response to a non-native image of an application program written for a non-native instruction set provides a native instruction or a native instruction routine. The run-time system collects profile data in response to execution of the native instructions to determine execution characteristics of the non-native instruction. Thereafter, the non-native instructions and the profile statistics are fed to a binary translator operating in a background mode and which is responsive to the profile data generated by the run-time system to form a translated native image. The run-time system and the binary translator are under the control of a server process. The non-native image is executed in two different environments with first portion executed as an interpreted image and remaining portions as a translated image. The run-time system includes an interpreter which is capable of handling condition codes corresponding to the non-native architecture. A technique is also provided to jacket calls between the two execution environments and to support object based services. Preferred techniques are also provide to determine interprocedural translation units. Further, intermixed translation/optimization techniques are discussed.

4 Claims, 79 Drawing Sheets